Scalable Gaussian Processes with Billions of Inducing Inputs via Tensor Train Decomposition

Pavel Izmailov[1] Alexander Novikov[2,3] Dmitry Kropotov[4]

[1]Cornell University

[2]National Research University Higher School of Economics

[3]Institute of Numerical Mathematics RAS

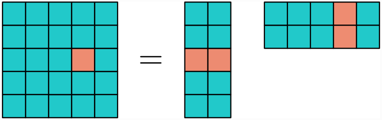
[4]Lomonosov Moscow State University

April 10, 2018

Tensor Train Decomposition [Oseledets 2011]

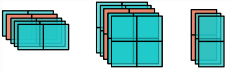
- Generalizes low rank approximation

Low-Rank



$$A_{3,4} = u_3^T \quad v_4$$

Tensor Train



$$B_{2,3,1} = u_2^T \quad v_3 \quad w_1$$

- Doesn't suffer from curse of dimensionality
- Allows fast implementation of linear algebra operations

ML Applications of TT

- TensorNet: DNN compression
  - Feed Forward [Novikov et al. 2015]
  - Convolutional [Garipov et al. 2016]
  - Recurrent [Yu et al. 2018]
- Markov Random Fields [Novikov et al. 2014]
- Theoretical analysis of RNN expressive power [Khrulkov et al. 2018]
- Discrete VAE [coming soon]

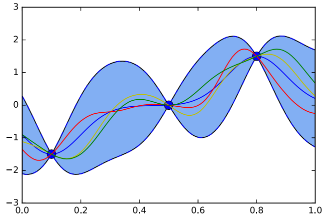
ML Applications of TT

- TensorNet: DNN compression
  - Feed Forward [Novikov et al. 2015]
  - Convolutional [Garipov et al. 2016]
  - Recurrent [Yu et al. 2018]
- Markov Random Fields [Novikov et al. 2014]
- Theoretical analysis of RNN expressive power [Khrulkov et al. 2018]
- Discrete VAE [coming soon]
- *TT-GP – Scalable GP framework*

Gaussian Processes

Definition

Gaussian process is a collection of random variables, any finite number of which have joint Gaussian distribution.

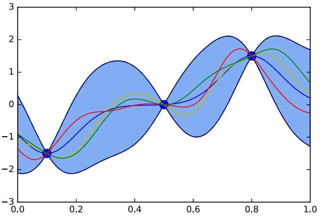


Posterior distribution of a one-dimensional Gaussian process

Gaussian Processes

Definition

Gaussian process is a collection of random variables, any finite number of which have joint Gaussian distribution.



Posterior distribution of a one-dimensional Gaussian process

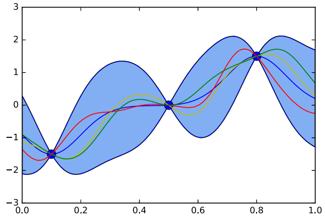
In Machine Learning GPs

- Allow automatic tunning of model complexity (non-parametric model)
- Provide principled uncertainty estimates
- Can discover complex non-linear patterns in data

Gaussian Processes

Definition

Gaussian process is a collection of random variables, any finite number of which have joint Gaussian distribution.

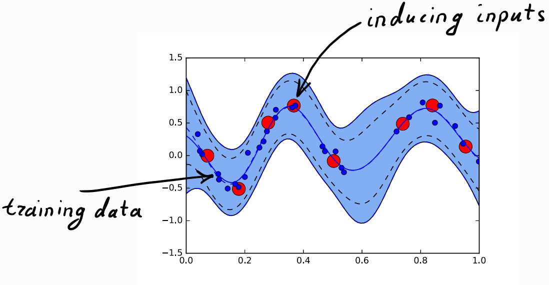


Posterior distribution of a one-dimensional Gaussian process

In Machine Learning GPs

- Allow automatic tunning of model complexity (non-parametric model)
- Provide principled uncertainty estimates
- Can discover complex non-linear patterns in data
- Exact inference is $\mathcal{O}(n^3)$

Inducing Inputs



Approximate posterior distribution based on inducing inputs

- Auxiliary observations that approximate the data
- Allow fast approximate inference

Previous Methods

- Classical methods [e.g. Snelson and Ghahramani 2005, Titsias 2009, Hensman et al. 2013] require $\mathcal{O}(nm^2 + m^3)$ computations, m is the number of inducing points
  - Applicable for large n (e.g. 10^6)
  - Infeasible for large $m \gg 10^3$

Previous Methods

- Classical methods [e.g. Snelson and Ghahramani 2005, Titsias 2009, Hensman et al. 2013] require $\mathcal{O}(nm^2 + m^3)$ computations, m is the number of inducing points
  - Applicable for large n (e.g. 10^6)
  - Infeasible for large $m \gg 10^3$
- KISS-GP [Wilson and Nickisch 2015] leverages the structure in the covariance matrices; requires $\mathcal{O}(n + m \log m)$ computations, $m = m_0^D$ and D is the number of features
  - Applicable for large n (e.g. 10^6) and m (e.g. 10^4)
  - Infeasible for large $D \gg 4$

Previous Methods

- Classical methods [e.g. Snelson and Ghahramani 2005, Titsias 2009, Hensman et al. 2013] require $\mathcal{O}(nm^2 + m^3)$ computations, m is the number of inducing points
  - Applicable for large n (e.g. 10^6)
  - Infeasible for large $m \gg 10^3$
- KISS-GP [Wilson and Nickisch 2015] leverages the structure in the covariance matrices; requires $\mathcal{O}(n + m \log m)$ computations, $m = m_0^D$ and D is the number of features
  - Applicable for large n (e.g. 10^6) and m (e.g. 10^4)
  - Infeasible for large $D \gg 4$
- *Tensor Train GP (TT-GP)* extends KISS-GP to high-dimensional problems
  - Applicable for large n (e.g. 10^6) and m (e.g. 10^8)
  - Applicable for larger D (e.g. 10)

ELBO [Hensman et al. 2013]

Evidence Lower Bound (ELBO) for GP regression:

$$\log p(y) \geq \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | k_i^T K_{mm}^{-1} \mu, \sigma^2) - \frac{1}{2\sigma^2} \left(\tilde{K}_{ii} + \text{tr}(k_i^T K_{mm}^{-1} \Sigma K_{mm}^{-1} k_i) \right) \right) -$$

$$\frac{1}{2} \left(\log \frac{|K_{mm}|}{|\Sigma|} - m + \text{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right) \to \max_{\mu, \Sigma, \theta, \sigma}$$

where

- $K_{mm} \in \mathbb{R}^{m \times m}$ is the covariance matrix computed at the inducing points
- $k_i \in \mathbb{R}^m$ is the vector of covariances between the i-th training object and the inducing points
- σ^2 is the noise variance
- $\mu \in \mathbb{R}^m$, $\Sigma \in \mathbb{R}^{m \times m}$ — variational parameters
- $\tilde{K}_{ii} = \delta^2 - k_i^T K_{mm}^{-1} k_i$, where δ^2 is the prior variance of the process at any point
- θ represents kernel hyper-parameters

ELBO

Assume m is very large (e.g. 10^{10})

$$\log p(y) \geq \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | k_i^T K_{mm}^{-1} \mu, \sigma^2) - \frac{1}{2\sigma^2} \left(\tilde{K}_{ii} + \text{tr}(k_i^T K_{mm}^{-1} \Sigma K_{mm}^{-1} k_i) \right) \right) -$$

$$\frac{1}{2} \left(\log \frac{|K_{mm}|}{|\Sigma|} - m + \text{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right)$$

ELBO + KISS-GP [Wilson and Nickisch 2015]

Assume m is very large (e.g. 10^{10})

$$\log p(y) \geq \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | w_i^T \mu, \sigma^2) - \frac{1}{2\sigma^2} \left(\tilde{K}_{ii} + \mathrm{tr}(w_i^T \Sigma w_i) \right) \right)$$
$$- \frac{1}{2} \left(\log \frac{|K_{mm}|}{|\Sigma|} - m + \mathrm{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right)$$

- Set inducing points on a grid
- Assume product kernel
- K_{mm} is in Kronecker product format
- $k_i \approx K_{mm} w_i$, w_i in Kronecker product format

TT-GP (Our Method)

$$\log p(y) \geq \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | w_i^T \mu, \sigma^2) - \frac{1}{2\sigma^2} \left(\tilde{K}_{ii} + \text{tr}(w_i^T \Sigma w_i) \right) \right)$$

$$- \frac{1}{2} \left(\log \frac{|K_{mm}|}{|\Sigma|} - m + \text{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right)$$

Restrict the format of variational parameters:

TT-GP (Our Method)

$$\log p(y) \geq \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | w_i^T \mu, \sigma^2) - \frac{1}{2\sigma^2} \left(\tilde{K}_{ii} + \mathrm{tr}(w_i^T \Sigma w_i) \right) \right)$$
$$- \frac{1}{2} \left(\log \frac{|K_{mm}|}{|\Sigma|} - m + \mathrm{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right)$$

Restrict the format of variational parameters:

- Σ in Kronecker product format

$$\Sigma = \Sigma^1 \otimes \Sigma^2 \otimes \ldots \otimes \Sigma^D$$

TT-GP (Our Method)

$$\log p(y) \geq \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | w_i^T \mu, \sigma^2) - \frac{1}{2\sigma^2} \left(\tilde{K}_{ii} + \text{tr}(w_i^T \Sigma w_i) \right) \right)$$
$$- \frac{1}{2} \left(\log \frac{|K_{mm}|}{|\Sigma|} - m + \text{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right)$$

Restrict the format of variational parameters:

- Σ in Kronecker product format

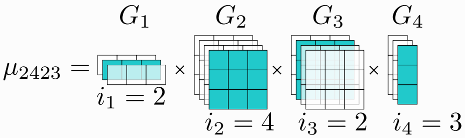
$$\Sigma = \Sigma^1 \otimes \Sigma^2 \otimes \ldots \otimes \Sigma^D$$

- μ in TT format
  - μ naturally reshapes to a tensor

Tensor Train format [Oseledets 2011]

Tensor μ is said to be represented in TT format if:

$$\mu(i_1, \dots, i_D) = \underbrace{G_1[i_1]}_{1\times r} \underbrace{G_2[i_2]}_{r\times r} \cdots \underbrace{G_D[i_D]}_{r\times 1}, \quad i_k \in \{1, \dots, m_0\}$$



$\boldsymbol{G}_k$ — TT-cores, $\quad r$ — TT-rank

- TT-format uses $\mathcal{O}\left(Dm_0r^2\right)$ memory to approximate a tensor with m_0^D elements
- Allows efficient implementation of linear algebra operations
- Generalizes Kronecker product format ($r = 1$)

TT-GP method

- Set inducing points Z on a grid in the feature space.
- Σ in Kronecker product format, μ in TT format
- Maximize the ELBO wrt to
  - TT-cores of μ
  - Kronecker factors of Σ
  - kernel hyper-parameters

Properties of TT-GP

- Computational complexity

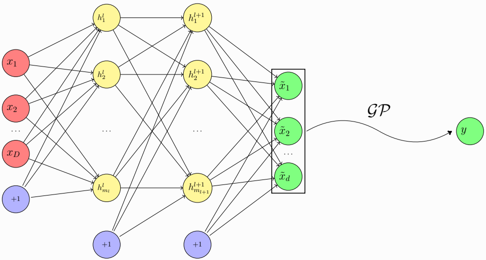
$$\mathcal{O}(nDm^{1/D}r^2 + Dm^{1/D}r^3 + Dm^{3/D});$$

  $m = m_0^D$, TT-ranks are on the scale of $r \approx 10$;

- In the experiments we use up to $n \approx 10^6$, $m \approx 10^{10}$

- Computationally tractable for large D
  - For $D >> 10$ more practical to train embedding

Deep Kernel Embedding [Wilson et al. 2016]



Given base kernel k, e.g. RBF

$$k(x, x') = \alpha^2 \cdot \exp(-\|x - x'\|^2/\beta^2),$$

define deep kernel as

$$k_{\text{net}}(x, x') = k(\text{net}(x), \text{net}(x')),$$

where k is the base kernel, net is a mapping performed by a DNN.

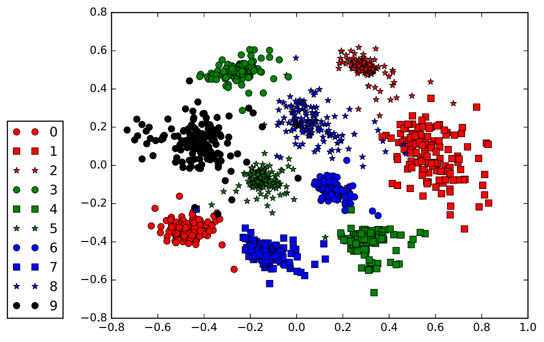
- DNN weights $\rightarrow$ kernel hyperparameters
- Train as before

Experiments: RBF kernel

Dataset			SVI-GP / KLSP-GP			TT-GP (Ours)			
Name	n	D	acc.	m	t (s)	acc.	m	d	t (s)
Powerplant	7654	4	0.94	200	10	**0.95**	35^4	-	5
Protein	36584	9	0.50	200	45	**0.56**	30^9	-	40
YearPred	$463K$	90	0.30	1000	597	**0.32**	10^6	6	105
Airline	$6M$	8	0.665^*	-	-	**0.694**	20^8	-	5200
svmguide1	3089	4	0.967	200	4	**0.969**	20^4	-	1
EEG	11984	14	**0.915**	1000	18	0.908	12^{10}	10	10
covtype bin	465K	54	0.817	1000	320	**0.852**	10^6	6	172

- SVI-GP – [Hensman et al. 2013]
- KLSP-GP – [Hensman et al. 2015]

Experiments: Deep Kernel Embedding



Learned representation for the Digits dataset, $n = 1797$, $D = 64$

Experiments: Deep kernels

Dataset		SV-DKL	DNN		TT-GP		
Name	n	acc.	acc.	t (s)	acc.	d	t (s)
Airline	$6M$	0.781	0.780	1055	$\mathbf{0.788 \pm 0.002}$	2	1375
CIFAR-10	$50K$	$-$	$\mathbf{0.915}$	166	0.908 ± 0.003	9	220
MNIST	$60K$	$-$	0.993	23	$\mathbf{0.9936 \pm 0.0004}$	10	64

- SV-DKL — [Wilson et al. 2016]

Discussion

TT-GP

- Uses Tensor Train decomposition and Kronecker format for variational parameters
- Scales to large n, m, D
- Naturally allows training deep kernels

Discussion

TT-GP

- Uses Tensor Train decomposition and Kronecker format for variational parameters
- Scales to large n, m, D
- Naturally allows training deep kernels
- Tends to overestimate uncertanties